Suggested Course Descriptions and Delivery Model

Year 1

Computer Programming I/A - Semester. An introduction to computer programming/software engineering and applications. The course introduces students to the fundamentals of computer programming, to simple control and data structures, to basic operating system commands, and to the use of text files. Students will learn to design, code, and test their own programs. Students will also apply mathematical skills throughout the course. It is recommended that teachers use the Scheme system for teaching this first semester of computer Programming. A skill certification exam is not available for this one semester course - see below.

Prerequisites: Algebra I, keyboarding proficiency, and computer technology.

Computer Programming I/B - Semester. An intermediate class in computer programming/software engineering and applications. Reviews and builds on the concepts introduced in CPI/A. Introduces students to more complex data structures and their uses, including sequential files, arrays, classes, and recursive processes. Students will learn to create more powerful programs.

Prerequisites: Algebra I, keyboarding proficiency, computer technology, and successful completion of CPI/A.

Note: Computer Programming I/A and Computer Programming I/B can be combined and offered as a full year course.

Year 2

Computer Programming II - Full year. An advanced class in computer programming/software engineering and applications. Reviews and builds on the concepts introduced in CPI-B. Introduces students to dynamic allocation of data, advanced utilization of classes, advanced GUI techniques, and advanced applications of recursion.

Prerequisite: Algebra I, keyboarding proficiency, computer technology, and successful completion of CPI/B.

Year 3

Computer Programming Individual Projects - Full Year. (Use Computer Programming II CIP Code). An advanced class in computer programming/software engineering where students can concentrate on their area of interest as directed by the instructor.

Computer Programming IA

Grade Levels: 10-12
Units of Credit: .5
CIP Code: 11.0201
Prerequisites: Algebra I, Keyboarding Proficiency, and Computer Technology (Computer Literary)
Skill Certification Exam: #820 Computer Programming 1A

COURSE DESCRIPTION

An introductory course in computer programming/software engineering and applications. The course introduces students to the fundamentals of computer programming. Students will learn to design, code, and test their own programs while applying mathematical concepts. It is recommended that teachers introduce programming concepts and problem solving skills to beginning programming students through high interest programs such as Lego robots, Alice, Java Script or KPL followed by an introduction to C++, Visual Basic or Java.

COURSE STANDARDS AND OBJECTIVES

STANDARD
110201-01 Students will be familiar with and use a programming environment.

OBJECTIVES

110201-0101 Demonstrate knowledge of external and internal computer hardware.

- Describe the functions of basic computer hardware devices (monitor, system board, printer, CD-ROM drive, hard drive, floppy drive, keyboard, mouse, adapters, ports, other devices).
- Describe the functions of the internal components of computers (CPU, RAM, ROM, registers, ALU).
- Translate to and from binary code (Computer number systems).

110201-0102 Demonstrate knowledge of software concepts.

- Define computer software.
- Explain the process of software installation.

110201-0103 Develop the ability to use a current operating system.

- Demonstrate how to load, save and back up files.
- Demonstrate how to rename and delete files.
- Demonstrate how to move, copy and compress files.
- Demonstrate how to display and print files.
- Demonstrate the ability to manage files on a PC and network
- Create Folders
    - Create and use appropriate directory and path structures
    - Copy files between folders
    - Understand the organization of files on a hard drive and a network
- Demonstrate how to execute a program.

110201-0104 Demonstrate the ability to use the editor to enter programs.

- Demonstrate how to enter text and commands.
- Demonstrate the process of selecting a block of text.
- Demonstrate how to move blocks of text.
- Demonstrate how to copy blocks of text.
- Demonstrate how to delete blocks of text.

110201-0105 Demonstrate the ability to compile, debug and execute programs.

- Demonstrate how to use the editor to compile and run programs.
- Understand the difference between syntax, run-time, and logic errors.
- Demonstrate how to debug programs.

STANDARD
110201-02 Students will employ accepted programming methodology.

OBJECTIVES;
110201-0201 Demonstrate the ability to use good programming style.

- Demonstrate how to use white space properly.
- Employ the use of case-sensitive commands for clarity.
- Construct programs with meaningful identifiers.

110201-0202 Employ the proper steps to programming in order.

- Prepare specifications for computer programs.
- Design solutions using algorithms such as flow charts, pseudocode, UML, N-S charts.
- Develop the code for a program.
- Test programs for effectiveness and completeness.
- Provide full documentation for a program.

110201-0203 Employ proper program design process.

- Use step-wise refinement (top-down design) in programming.
- Employ program modularity in writing programs.
- Produce logical algorithms to solve problems with a computer program.

110201-0204 Demonstrate the ability to program for automatic error checking (robustness).

- Explain how to protect program execution from incorrect input.
- Describe how to protect program execution from run-time errors.
- Employ verification to protect program results from logic errors.

STANDARD
110201-03 Students will properly use language-fundamental commands and operations.

OBJECTIVES:
110201-0301 Demonstrate the ability to use basic elements of a specific language.

- Write programs using a language-specific template.
- Declare and assign values to constants and variables in programs.
- Output text with formatting.
- Demonstrate how to pause programs in order to view the output.
- Demonstrate the ability to use input/output commands in programs.
- Input values into identifiers.

- Output values stored in identifiers.

110201-0302 Employ arithmetic expression in programs.
- Arithmetic operators.
- Unary operators.
- Modulus division.
- Integer division.
- Order of operators.

110201-0303 Demonstrate the ability to use atomic data types in programs.
- Declare and use integer identifiers.
- Declare and use character identifiers.
- Declare and use floating point (real) identifiers.
- Declare and use Boolean identifiers.
- Declare and use constants.

110201-0304 Demonstrate the ability to use strings in programs.
- Declare string identifier.
- Input string identifiers.
- Output string identifiers.

STANDARD
110201-04 Students will properly employ control structures.

OBJECTIVES:
110201-0401 Demonstrate the ability to use relational and logical operators in programs.
- Compare values using relational operators.
- Form complex expressions using logical operators.

110201-0402 Demonstrate the ability to use decisions in programs.
- Employ simple IF structures.
- Use IF-ELSE structures.
- Write programs with nested IF-ELSE structures.
- Make multiple-way selections (switch, case).

110201-0403 Demonstrate the ability to use loops in programs.
- Use initial, terminal, and incremental values in loops.
- Construct both pre-test and post-test loops.
- Demonstrate how to use counted loops.
- Describe the use of flagged (sentinel-controlled) loops.
- Utilize nested loops.
- Explain how to avoid infinite loops.
- Accumulate running totals using loops.

110201-0404 Demonstrate the ability to use sub-programs in programs.
- Demonstrate how to use language-predefined sub-programs.
- Call sub-programs.
- Develop sub-programs.
- Send to and retrieve data from sub-programs.

- Utilize value, constant, and reference parameters.
- Understand the scope of identifiers in sub-programs.

STANDARD

110201-05 Students will demonstrate knowledge of current ethical issues dealing with computers and information in society.

OBJECTIVES;

110201-0501 Demonstrate knowledge of programmer ethics.

- Show knowledge of the importance of correct programming.
- Check for program correctness using verification.
- Know proper and improper standards in programming.

110201-0502 Demonstrate knowledge of the social and ethical consequences of computers.

- Describe how computer-controlled automation affects workers and management.
- Explain the ramifications of society's dependance on computers.
- Identify advantages and disadvantages of changing workplace environments.

110201-0503 Demonstrate knowledge of the right to privacy.

- Explain how computers can compromise privacy.
- Exhibit knowledge of privacy laws.
- Describe responsibilities of people who control computer information.

110201-0504 Demonstrate knowledge of computer, information and software security.

- Exhibit knowledge of copyright laws.
- Explain how computers can be used to compromise copyright laws.
- Give examples of ways to protect information on computer systems.
- Identify ways to protect against computer viruses.

STANDARD

110201-06 Students will develop an awareness of career opportunities in the Computer Programming/Software Engineering industry and of its history.

OBJECTIVES

110201-0601 Develop career awareness related to working in the Computer Programming/Software Engineering industry.

- Identify personal interests and abilities related to Computer Programming/Software Engineering careers
  - Identify personal creative talents
  - Identify technical/programming talents
  - Identify organizational and leadership skills
  - Explore aptitude for innovation
  - Determine aptitude for working as a member of a Computer Programming/Software Engineering team
- Identify Computer Science career fields
  - Software Engineer
  - Systems Analyst
  - Applications Programmer (Gaming, Multimedia Etc.)

- Investigate career opportunities, trends, and requirements related to Computer Programming/Software Engineering careers
  - Identify the members of a Computer Programming/Software Engineering team: Team Leader, Analyst, Sr. Developer, Jr. Developer, and Client/Subject Matter Expert
  - Describe work performed by each member of the Computer Programming/Software Engineering team
  - Investigate trends associated with Computer Programming/Software Engineering careers
  - Develop a realistic Student Education Occupation Plan (SEOP) to help guide further educational pursuits
- Identify factors for employability and advancement in Computer Programming/Software Engineering careers
  - Survey existing Computer Programming/Software Engineering businesses to determine what training is required
  - Survey universities and colleges to determine higher education options
  - Develop employability competencies/characteristics: responsibility, dependability, respect, and cooperation
  - Achieve high standards of personal performance
  - Develop a positive work ethic
  - Compile a portfolio of the individual and group programs developed during the course
- Discuss relevant history of computer technology and the Computer Programming/Software Engineering industry.

Computer Programming IB

Grade Levels: 10-12
Units of Credit: .5
CIP Code: 11.0201
Prerequisites: Computer Programming IA, Algebra I, Keyboarding Proficiency, and Computer Technology (Computer Literary)
Skill Certification Exam: 822 C++, 824 Java, 826 VB

STANDARD
110201-07 Students will employ proper static data structures.

OBJECTIVES:
110201-0701 Demonstrate the ability to use static arrays in programs.
- Declare arrays all applicable types.
- Initialize arrays.
- Input data into arrays.
- Output data from arrays.
- Perform operations on arrays.
- Perform sequential searches on arrays.
- Perform a sort on an array.

110201-0702 Utilize multidimensional arrays.
- Initialize arrays.
- Input data into arrays.
- Output data from arrays.
- Perform operations on arrays.
- Perform searches on arrays.

110201-0703 Demonstrate the ability to use strings in programs.
- Compare string identifiers.
- Find the length of a string.
- Copy part or all of string identifiers into other strings.
- Concatenate string identifiers.
- Locate and delete sub-string positions.
- Insert strings into other strings.

STANDARD
110201-08 Students will properly employ object-oriented programming techniques.

OBJECTIVES;
110201-0801 Demonstrate the ability to use classes.
- Use objects.
- Use object data members.
- Use object member functions (methods).

110201-0802 Demonstrate the ability to create and use user-defined classes.
- Create and use user-defined data members.

- Create and use user-defined methods.
- Create and use overloaded operators (C++).

STANDARD
110201-09 Students will properly use sequential files.

OBJECTIVES:
110201-0901 Demonstrate the ability to use sequential files in programs.

- Create and initialize sequential files.
- Store data to sequential files.
- Retrieve data from sequential files.
- Update sequential files.

STANDARD
110201-10 Students will apply appropriate programming skill as an effective member of a team.

OBJECTIVES;
110201-1001 Demonstrate the ability to apply knowledge to a programming project.

- Formalize specifications.
- Choose proper input parameters.
- Choose appropriate data structures and processing.
- Design appropriate output.
- Use appropriate test data.
- Write good documentation.

110201-1002 Demonstrate the ability to use teamwork and collaboration in a programming project.

- Divide a project among programmers.
- Present work to a group.
- Coordinate work with others in the group.
- Complete assigned work according to predetermined deadlines.
- Participate in a peer performance evaluation.
- Demonstrate professionalism in team relationships, communication, timeliness, and attitude.

Computer Programming II

Grade Levels: 10-12
Units of Credit: 1
CIP Code: 11.0202
Prerequisites: Algebra I, Keyboarding Proficiency, Computer Technology and Computer Programming IA & B
Skill Certification Exam: #803 C++, 835 Java, and/or AP Computer Science

COURSE DESCRIPTION

An advanced course in computer programming/software engineering and applications. Reviews and builds on the concepts introduced in CP I. Introduces students to dynamic allocation of data, advanced utilization of classes, advanced GUI techniques, and advanced applications of recursion through the application of mathematical concepts.

CORE STANDARDS, OBJECTIVES, AND INDICATORS

STANDARD
110202-01 Students will develop applications which make advanced use of the skills and concepts developed in Computer Programming I.

OBJECTIVES:
110202-0101 Demonstrate the ability to develop advanced applications.
- Develop advanced applications using input, calculations, and output.
- Develop advanced applications using IF structures.
- Develop advanced applications using iteration.
- Develop advanced applications using sub-programs.
- Develop advanced applications in object-oriented programming.
- Develop advanced applications using recursion.
- Develop advanced applications using arrays.
- Develop advanced application projects.
- Develop advanced applications using files (sequential files) or a simple database.

STANDARD
110202-02 Students will use more efficient searching and sorting algorithms.

OBJECTIVES::
110202-0201 Demonstrate the ability to search data structures in programs.
- Develop a binary search.
- Develop a hash search including best and worst, average and hash searches.
- Compare efficiency of sequential and binary searches.

110202-0202 Demonstrate the ability to sort data structures in programs.
- Sort arrays using quadratic (n^2) sorts.
- Sort arrays using binary (n log n) sorts.
- Compare the efficiency of the various sorts using BigO notation including best, worst and average.

STANDARD
110202-03 Students will implement and manipulate a simple database.

OBJECTIVES::
110202-0301 Demonstrate the ability to use random access files in programs.
- Create and initialize random access files.
- Store data to random access files.
- Read data from random access files.
- Update random access files.
- Index random access files.
- Utilize hashing on random access files.

STANDARD
110202-04 Students will properly employ dynamic data structures and abstract data types (ADTs).

OBJECTIVES
110202-0401 Demonstrate the ability to use linked lists in programs.
- Declare pointer identifiers.
- Create node identifiers.
- Insert nodes into a linked list (front, middle, end).
- Delete nodes from a linked list (front, middle, end).
- Output the values in a linked list.
- Search for a value in a linked list.
- Use header and non-header linked lists.
- Perform other linked list operations.
- Develop linked list applications.

110202-0402 Demonstrate the ability to use stacks (arrays and linked lists) in programs.
- Declare stack structures.
- Initialize stacks.
- Check for empty and full stacks.
- Push on to and pop off values from stacks.
- Develop stack applications.

110202-0403 Demonstrate the ability to use queues (arrays and linked lists) in programs.
- Declare queue structures.
- Check for empty and full queues.
- Initialize queues.
- Enqueue values on to and dequeue values off of queues.
- Develop queue applications.

110202-0404 Demonstrate the ability to use binary trees in programs.
- Declare pointer identifiers.
- Create binary tree nodes identifiers.
- Insert nodes into a binary tree.
- Delete nodes from a binary tree.
- Output the values in a binary tree.
- Search for values in ordered binary trees.

- Develop binary tree applications.

STANDARD
110202-05 Students will design and implement classes using inheritance and composition.

OBJECTIVES
110202-0501 Create user-defined inherited classes.

110202-0502 Demonstrate overloading techniques.
- Demonstrate function overloading
- Demonstrate operator overloading (C++ only)

STANDARD
110202-06 Students will develop an individual program of significant complexity and size (300-500 lines) and portfolio of their work.

OBJECTIVES
110202-0601 Create an individual program of significant complexity and size (300-500 lines).

110202-0602 Compile a portfolio of the individual and group programs developed.

STANDARD
110202-07 Students will participate in a work-based learning experience and/or competition.

OBJECTIVES
110202-0701 Participate in a work-based learning experience.
- Field trip to a software engineering firm
- Job shadow
- Internship
- Industry guest speaker
- Post-secondary guest speaker
- Industry interview

110202-0702 Participate at a student programming competition.
- University of Utah High School Computer Programming Contest
- Utah State University High School Computer Programming Contest
- Utah Valley State College Technology Fair (robotics)
- FBLA Programming Contest
- SkillsUSA Programming Contest